JPRS 77092

2 January 1981

China Report

SCIENCE AND TECHNOLOGY

No. 75

FBIS FOREIGN BROADCAST INFORMATION SERVICE

NOTE

JPRS publications contain information primarily from foreign newspapers, periodicals and books, but also from news agency transmissions and broadcasts. Materials from foreign-language sources are translated; those from English-language sources are transcribed or reprinted, with the original phrasing and other characteristics retained.

Headlines, editorial reports, and material enclosed in brackets [] are supplied by JPRS. Processing indicators such as [Text] or [Excerpt] in the first line of each item, or following the last line of a brief, indicate how the original information was processed. Where no processing indicator is given, the information was summarized or extracted.

Unfamiliar names rendered phonetically or transliterated are enclosed in parentheses. Words or names preceded by a question mark and enclosed in parentheses were not clear in the original but have been supplied as appropriate in context. Other unattributed parenthetical notes within the body of an item originate with the source. Times within items are as given by source.

The contents of this publication in no way represent the policies, views or attitudes of the U.S. Government.

PROCUREMENT OF PUBLICATIONS

JPRS publications may be ordered from the National Technical Information Service, Springfield, Virginia 22161. In ordering, it is recommended that the JPRS number, title, date and author, if applicable, of publication be cited.

Current JPRS publications are announced in Government Reports Announcements issued semi-monthly by the National Technical Information Service, and are listed in the Monthly Catalog of U.S. Government Publications issued by the Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.

Indexes to this report (by keyword, author, personal names, title and series) are available from Bell & Howell, Old Mansfield Road, Wooster, Ohio 44691.

Correspondence pertaining to matters other than procurement may be addressed to Joint Publications Research Service, 1000 North Glebe Road, Arlington, Virginia 22201.

19277:91

REPORT DOCUMENTATION PAGE	1. REPORT NO. JPRS 77092	2.	3. Recipient's Accession No.
4. Title and Subtitle: CHINA REPORT: SCIENCE AND TECHNOLOGY, No. 75			5. Report Date: 2 January 1981
			6.
7. Author(s)			8. Performing Organization Rept. No.
9. Performing Organization Name and Address: Joint Publications Research Service, 1000 North Glebe Road, Arlington, Virginia 22201			10. Project/Task/Work Unit No.
			11. Contract(C) or Grant(G) No. (C) (G)
12. Sponsoring Organization Name and Address: As above			13. Type of Report & Period Covered
			14.

15. Supplementary Notes

16. Abstract (Limit: 200 words)

This serial report contains articles, abstracts and news items on national developments in science and technology; and physical, applied and life sciences in China.

17. Document Analysis a. Descriptors

CHINA
National Developments
Chinese Academy of Sciences
Physical Sciences
Applied Sciences
Life Sciences

b. Identifiers/Open-Ended Terms

c. COSATI Field/Group 01, 02, 04, 06, 07, 08, 09, 11, 12, 13, 20

18. Availability Statement: Unlimited Availability, Sold by NTIS, Springfield, Virginia 22161	19. Security Class (This Report) UNCLASSIFIED	21. No. of Pages 49
	20. Security Class (This Page) UNCLASSIFIED	22. Price

JPRS 77092

2 January 1981

CHINA REPORT

SCIENCE AND TECHNOLOGY

No. 75

CONTENTS

APPLIED SCIENCES

Design of Diagnostic System of 151-2 Computer Described
(Li Yanhong; ZHONGXIAOXING JISUANJI, 25 Jul 80) 1

303-1 General Wage Program Introduced
(Chen Shengdong; DIANZI KEXUE JISHU, 4 Oct 80) 38

DJS130-PZ5 Interface Operation Detailed
(Fu Shouchun, Gu Haoran; DIANZIJISHU, 20 Mar 80) 44

APPLIED SCIENCES

DESIGN OF DIAGNOSTIC SYSTEM OF 151-2 COMPUTER DESCRIBED

Liaoning ZHONGXIAOXING JISUANJI [MINI-MICRO SYSTEMS] in Chinese No 3, 25 Jul 80 pp 1-16, 30

[Article by Li Yanhong [2621 1750 7703] of Changsha Engineering College: "Microdiagnostic System of the 151-2 Computer"]

[Text] Abstract

This article describes the design of the diagnostic system of the 151-2 computer, mainly the principle of microdiagnosis, the hardware core, the design of the diagnosis of microprograms, automatic test of the hardware core, method of generation of the test code and its advantages, diagnostic commands system and its function and mode of operation, basic program block diagram with special emphasis on pointing out the function of this system, the indicators of performance and its practical value in the diagnosis of medium and small computers.

I. Introduction

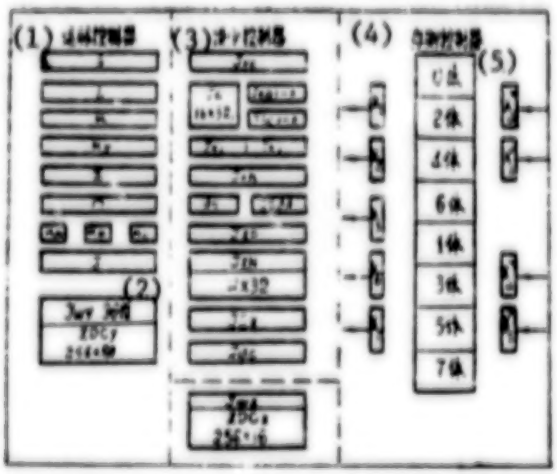
As integration of digital computers develops and the function and application of the machines expand, the demands upon the realiability and maintenance ability have become more and more urgent. Therefore, in recent years, techniques of diagnosis of computers have been studied in our nation and abroad to adapt to this situation. A relatively complete diagnostic system has been designed for the major parts of this computer.

The diagnostic means are determined by the type of malfunction (fault). They generally can be classified as frequently occurring fixed malfunctions and chance and intermittent malfunctions. The former can be further divided into simple malfunctions and multiple malfunctions. Also, diagnosis of fixed malfunctions is the main purpose of diagnosis and diagnosis of intermittent malfunctions auxiliary. Diagnosis of malfunctions is the main purpose of diagnosis and function audits are auxiliary. According to ordinary understanding, diagnosis is to divide the machine logically into several diagnostic regions (test regions). The control unit sends a test code to its input end and observes the response of that circuit at the output end, i.e., retrieving the test result and then by a comparison of the test result and the standard results, decide the area of the malfunction. Here, because the sources of the hardware and the software used in the process are not the same, the types and

effects of diagnosis will be very different. When the machine's hardware and commands are used to carry out diagnosis, we are using an audit program to carry out function audit. When special diagnostic equipment and special diagnostic commands are used to carry out diagnosis, we are using a diagnostic program to carry out diagnosis of malfunctions. This machine uses the latter and it is microdiagnosis, this means using a microprogram composed of a series of microcommands to diagnose malfunctions of a microchannel. This can be easily realized by microprogram controlled machines.

II. Block Diagram of the Processor

To understand the object of diagnosis, a simple block diagram of the processor is drawn for illustration.



Block Diagram of Processor

Key:
(1) Computation controller
(2) 96 digits
(3) Command controller
(4) Memory controller
(5) 0 part
2 part
4 part [etc]

(I) Total System Function Indicators

The main frequdncy is 2.5 MC. The speed is 600,000 times per second. There are two word lengths: The single word length is 32 bits, the double word length is 64 bits. The main memory has 65,00 words (which can be expanded to 260,000 words). Two systems are used. The memory combination is a dual system modulo 2 or a single system modulo 1. The operational cycle of the main memory is 1.2 us. There are 64 channels for peripheral equipment (half for special equipment and half for general purpose equipment). The computer has a relatively complete peripheral equipment. In particular, it has two kinds of peripheral equipment (narrow line printer and capacitance input device) which can be directly hooked onto the processor for diagnosis and debugging the machine.

It has a complete memory control unit, a command control unit and an operational control unit. The command control unit uses a definite degree of precontrol techniques. The memory control unit uses a time-shared queuing technique. The main memory is modular. The operational control unit is microprogrammed (standardized operational control).

(II) Structural Characteristics

The core unit multiplier (Π) of the arithmetic unit and the full adder (Σ) are connected in series as a unit. There is a bus switch. The results of computation are sent to the results register (J_{YC}) via the shift switch (K_Y) and the bus switch (K). There is a dual channel structure. The registers S, L (the first operational digit registers) and J (the second operational digit register) have a channel that directly passes through the bus switch. There is also a channel to the bus through the multiplier-full adder. The arithmetic unit also has an N counter unrelated to the bus switch, a Kn counter switch and some independent status triggers.

The operational control has a microprogram memory ZDC_Y, microcommands register J_{WY}, decoder YM_Y and some control conditions.

The main functional units of the command control unit are: entry command address counter JS_Z, command control unit's current-command address adder Q_Z, operational control unit's operational command address counter (break point) JS_{dd}, command queuing unit (analyzer(J_{ZP}, general purpose accumulator register J_L, results register J_{YC}, command buffer register J_{ZH}, request trigger and a counter that controls the entry and exit of commands.

The memory control unit has a queuing unit, address transformer, address switch and numerical code switch.

III. Principles of Diagnosis

(I) Mode of Operation

This diagnostic part is hidden inside the processor's diagnostic controller. It has an independent hardware and software system by which it performs diagnosis of the operational control unit. Therefore it is similar to the method of an internal "diagnosis by an outside machine." The actual process of operation is to generate the needed test codes and to record the programs and data on a data tape (punched paper tape, magnetic tape). At the time of diagnosis, the test codes are transferred into the main memory. After start-up the test code under the control of the diagnostic control unit is accessed and loaded onto the unit to be diagnosed via the sweeping unit. The test results are retrieved into the diagnostic unit and compared with the standard results accessed from the main memory. Finally, a decision is made as to correctness and thus a group of test results is obtained. When all the test groups in that test region have been tested, they are comprehensively analyzed. The fault is fixed accurately according to the diagnostic dictionary and the results are finally printed.

(II) Marking Out the Test Regions

The scope of the test regions depends upon the tested logic function block and determines the problem of determining the test codes.

The nature of marking out the test regions is marking out of the test channels. Therefore marking out the regions is not a random division of the entire logic unit because if so, some unnecessary test channels will surely be added, and the purpose of testing the original logic channels may not even be fulfilled.

The order of marking out the test regions facilitates isolating the problems. Therefore, testing regions are ordered from the small to the large and from the inside to the outside. A smooth testing of the test regions in front provides the possibility of testing the test regions that follow. The latter relies upon the front region to pass on the test code. The latter is a newly added test channel, i.e., a new test region. In this way, the order of the test region reflects the snowball method of testing.

(III) The Method of Peripheral Equipment

In the process of loading the test code and transmitting and retrieving the test results, the original data channels must all be opened. The microaction signals needed here cannot be directly accessed from the operational control unit's microprogram memory because their arrangement does not suit the needs of microdiagnosis. Therefore, the method of adding equipment to the microcommand register must be used.

IV. Hardware System

(I) Hardware Block Diagram

K — Output bus switch of the operational control unit used here as a sweep-in unit;

J^{Q}_{yc} — Register of the output results of the command control unit, used here as the retrieving register and comparison register (containing comparison circuits);

J_{ZP} — Command queuing device of the command control unit, used here as the diagnostic commands register and the digital code buffer register.

JS_{Z} — Entry counter of the commands of the command control unit used here as the diagnostic commands counter;

J_{dc} — Stored data address register;

JS_{dd} — Breakpoint counter of the command control unit used here as a variable address register;

J_{K} — Switch register of the command control unit used here as the surface board switch number during autotest;

J_{st} Time counter of the command control unit used here as the entry address of the diagnostic read only memory during autotest;

JS_{ZHQ} Commands request number counter of the commands control unit used here as the peripherally installed operational control unit's microcommand word counter (32 digits per fixed equipment);

JS_{sr} Input commands' input message counter;

Q_Z Command control unit's current command address adder used here as the variable address adder;

ZDC_Z Diagnostic read only memory (microprogram memory);

J_{WZ} Diagnostic microcommand register;

K_{WZ} Diagnostic microcommand register's input switch;

YM_{WZ} Diagnostic command decoder;

K_{XT} Diagnostic microprogram selector switch;

KZ_{XT} Diagnostic microprogram selector control;

Kd_{WZ} Entry address switch of the diagnostic read only memory;

QD_{dWZ} Address drive of the diagnostic read only memory;

PL Decide "0" circuit;

XO Parity checking circuit;

C_{Q7} Access command request trigger of the command control unit used here as the request trigger for accessing commands and data;

C_{Q3} Stored data request trigger;

C_{gZ} Diagnostic process signal trigger;

C_{yg} Malfunction signal trigger for diagnosis;

C_Z Clocked diagnostic work cycle control trigger;

C_{ZY} Operational control command decoder trigger for diagnostic control;

C_{cd} Length signal trigger;

C_{PZ} Beat signal trigger used in the cycle of operational control commands;

C_{bZ} Polarity selection and control trigger of the diagnostic work clock;

C^{*}_{SR} Signal trigger for sweeping into the ith channel or the jth channel when testing the accumulator register;

C^{*}_{JSR} Signal trigger of the input commands to send data;

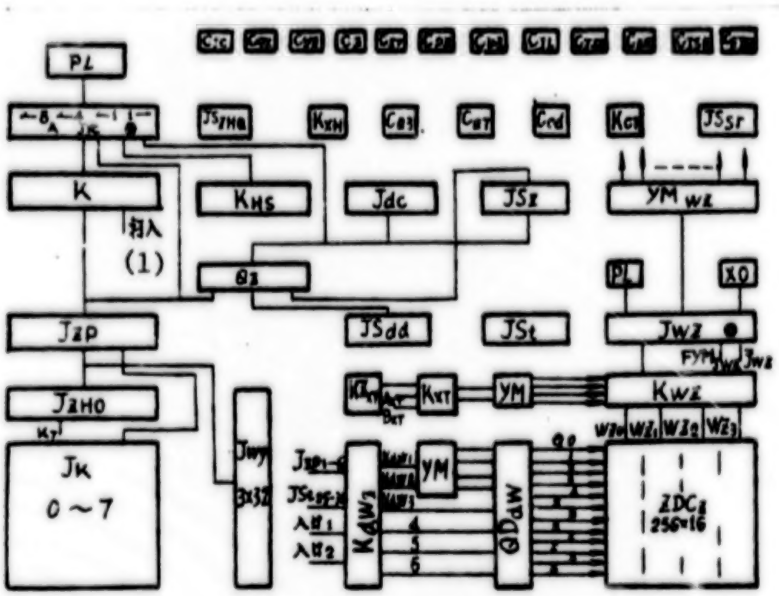
C^{*}_{JSD} End of input signal trigger;

C_{CT} End of printout signal trigger;

K_{XH} Wrong stop switch;

K Cycle switch;

Of these, except for the microprogram used as the main control of diagnosis, the majority also uses the main functional units of the command control unit. Here, the main diagnostic function units are J_{ZP} and $J_y{}^{Q}{}_{c}$. The former is a command control unit. The latter is a diagnostic analysis unit. The microprogram is a control unit of diagnostic operation. It is the memory's logic unit that executes the various microactions of the diagnostic commands.



Block diagram of hardware

Key: (1) Sweep in

(II) Design of the Diagnostic Microprogram

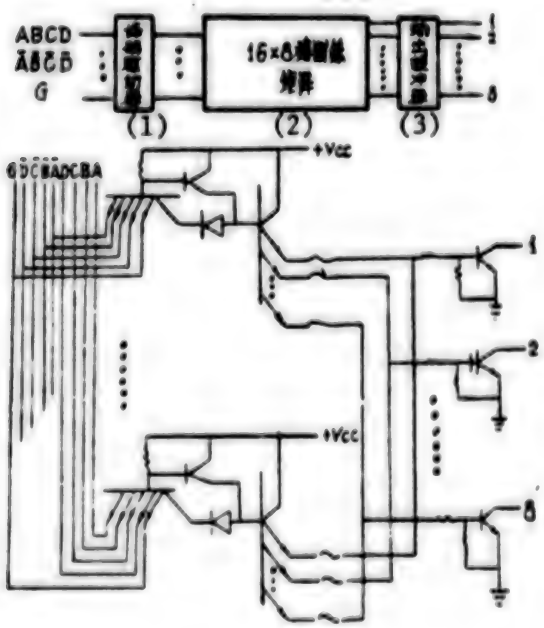
Since the technology of designing microprograms as a memory logic unit came into being, it has received popular attention in our nation and abroad because of its advantages. In foreign nations, this has already become reality. In our nation, it used to be something that could only be dreamed about. But since the successful trial development and popularization of the medium scale integrated circuit read only memory component (PROM) in our nation, small,

medium and large computers with microprogrammed controls have emerged one after the other. Although they are all controlled by microprograms, the emphasis is not the same when considering the designs of the microprograms. For small computers, speed is not the main problem, therefore the point of consideration is how to solve the problem concerning conflicts between the development of the advantages of the design of microprograms and improvement of the speed of the machine. Concerning this point, the 151-2 model's operational control microprogram design has already become an example. The design of the diagnostic microprogram supporting it must take into consideration the problem of speed and the problem of conservation. This is one of the guiding thoughts in the design of the diagnostic microprograms of this system.

1. Structure of the Control Unit of the Diagnostic Microprogram

(1) Microprogram Storage ZDC_Z

It consists of a 4 x 64 x 16 matrix, the 4 x 64 x 8 is an inserted unit. Thus two inserted boards form 256 words x 8 bits (sharing one board with the ZDC_y of the operational control unit). They are divided into four parts. Each part has 4 chips. Each chip has 16 words x 8 bits. These 8-bit messages are realized by a nickel-chromium fused filament (See component logic diagram). This is a four-part parallel read-out structure to facilitate increasing the machine's speed. For the 25 diagnostic commands and the autotest "program blocks," the 256-unit capacity will never be used up. This seems to be a waste. This is because the use of the standard board of the operational control unit reduces the backup boards needed but when the units not being used form 16 continuous words, these components will not have to be welded or can be welded but not etched to serve as a backup. In this way, there will be no waste. A word length of 16 bits is bearly sufficient to perform such microfunctions of so many commands as accessing commands, accessing data and performing diagnostic operations. This is precisely the problem that is characteristic of this microprogram design and that needs to be solved.



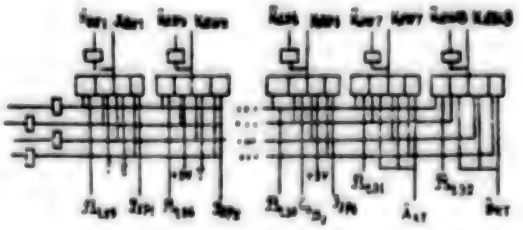
Key:
(1) Decoding drive
(2) 16 x 8 fuse matrix
(3) Output buffer

PROM components logic diagram

(2) Entry Address Switch K_{dWZ}

The method of addressing of the microprogram storage is chip selection (each chip), word selection and part selection. K_{dWZ} has 8 bits, $K_{dWZ1,2}$ is used for chip selection after decoding; K_{dWZ3-6} is used for word selection (the component contains a decoding drive) via the address drive inside the inserted unit; $K_{dWZ7,8}$ is used for part selection.

This switch has a total of four channels, i.e., four entry address sources: (1) entry address at time of autotest. It originates from the timer JS_t 25-32. This shares the JS_t of the command control unit and uses it as the hardware core of the hardware for automatic testing and diagnosis; (2) When the diagnostic program functions and operates the code channel, it uses the operation l codes of the commands as the addresses of the microcommands (not through encoding). They originate fr[illegible]perational codes of the J_{ZP} queuing device (actually, they originat[illegible] from the dead wait counter JS_{3d}, because it also stores operational code[illegible]nd in addition, the names of these components all retain the original calls [illegible]he command control unit, and diagnosis only shares them, the same in the following); (3) Entry 1 channel is the entry address into the diagnostic program after automatic tests of two groups of counters and after completion of automatic tests and after they are ready to be entered into the diagnostic program. These two groups of counters are diagnostic components and they also share the command control unit. Readiness for entry into the diagnostic program includes the "diagnostic program procedure" signal trigger C_{gZ}, and accessing of the surface board commands to the queuing device (register) J_{ZP}, and the commands on the surface board J_{K7} are the shut off commands or unconditional transfer instructions. When automatic tests end and the diagnostic program needs to be input, the shut off command is placed on J_{K7} and the input command is placed on J_{K0}. If the program has already been input into the main memory, the unconditional transfer commands are placed on J_{K0} (transferring to the entry of the diagnostic program). Any one of the two types of commands mentioned above can be placed on J_{K7} and any one can be transferred to the commands on J_{K0} because when automatic testing ends, the command counter JS_Z is cleared and set at "0", and the commands are transformed to the No 8 unit. J_{K0-7} occupies the No 8 to No 15 addresses of the main memory; (4) Entry 2 channel is the entry address for accessing commands and data. Since these functions are shared, a common entry is given. The logic of K_{dWZ} is illustrated below.



Entry switch Kd_{WZ} logic diagram (using standard inserted components to realize it)

(3) Microcommand Register J_{WZ}

It is a 16-bit register composed of the D trigger. The D trigger has a function of "supporting the top and opening the bottom." Its logic is shown in the following diagram.

J_{WZ} and its channel logic diagram

The input channels are:

(a) K_{WZ} channel is the microcommand read out channel;

(b) FYM_{WZ} channel is the channel of the automatic test decoder;

(c) BC channel is the retention channel but it is not used for cyclic use by the microcommands. It is set up to search for modulo 2 plus (search for the sum of the command codes;

(d) All clear "0" channel is the ZZ "0" J_{WZ}(R) channel set up so that debugging and normal operations will not be affected by diagnostic microactivity.

Support installations are:

(a) Decide "0" circuit is for deciding the codes and full "0" during automatic testing, i.e., for longitudinal parity check.

(b) Parity checking circuit is the horizontal parity check set up to monitor the correctness of the microprogram storage when the microprogram is in operation.

These checking circuits will immediately stop the machine when errors are discovered at different stages.

(4) Decoder YMZ

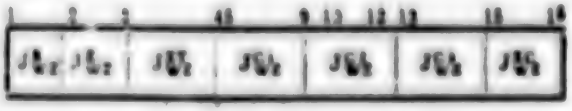
(a) Uncontrolled decoding;

(b) Controllable decoding to conserve the use of read only memory and it is even irreplaceable.

To facilitate testing and searching for the sum of the codes during automatic testing, all the standard inserted decoder units have a reverse decoding circuit and they must be able to retrieve all coded dots without hindrance. Therefore the control conditions are added onto the second level decoding circuit (forming an unstandardized control board).

2. Format of Microcommands and Division of Syllables

Because the standard read only memory inserted chip is used, the format of microcommands and division of syllables are necessarily limited in some ways.



Format of J_{WZ}

(1) Syllable Distribution

Many formats of microcommands of microprogram designs contain various types of characteristic digits, such as lower microcommand addresses, transfer condition codes, checking digit, and operational syllables etc, but this design has set up only two-digit characteristic signals according to the actual situation, two digit selector syllables and three operational syllables and a one-digit checking digit.

(2) Function of Each Syllable

(a) J^{R}_{WZ} is called the entry address signal digit. When the diagnostic program is in operation, $J^{R}_{WZ} = 1$. It signifies the operational code has been accessed as the entry address. $J^{R}_{WZ} = 0$ signifies that the entry 2 of a fixed microprogram has been accessed at the entry address.

(b) J^{Z}_{WZ} is called the transfer signal digit. $J^{Z}_{WR} = 1$ is unconditional transfer, i.e., the transfer condition is accessed and the selector address is decided according to the related selector logic.

(c) J_{WZ}^{XT} is called the selector syllable, totalling 2 digits called A_{XT}, B_{XT}.

(d) J_{WZ}^{C1} is called the first operational syllable, totalling 5 digits.

(e) J_{WZ}^{C2} is called the second operational syllable, totalling 3 digits.

(f) J_{WZ}^{C3} is called the third operational syllable, totalling 3 digits.

(g) J_{WZ}^{XD} is called the parity checking digit.

3. Selector Control

The course of the microprogram like ordinary programs, will encounter the problem of transfers of conditions. Therefore usual design of microprograms must solve the problem of transfer of conditions of the microcommands. There are the following methods:

(1) Setting Up Condition Coding Syllabes

Especially in medium and large computers, there are many conditions in the course of computation and condition coding syllables must be set up. At the time of start-up, a "call" is made according to the code number. If the called condition is satisfied, the selector's direction is thus decided by the related selector logic. Unconditional transfer are also given code numbers. They will select and access the part address on the address syllable as the selector address. This is the plan for the 151-2 computer's operational control microprogram to process transfer of conditions.

(2) Setting Up the Conditions Signal Digit

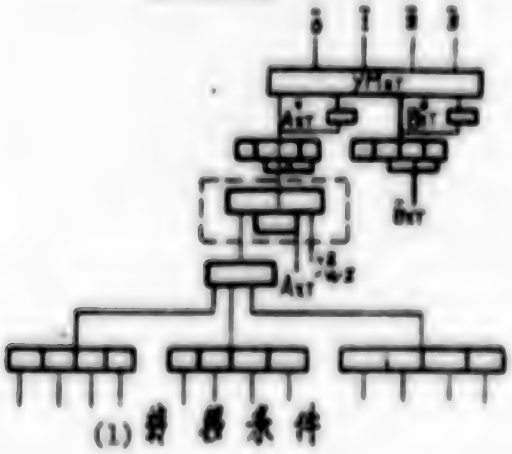
In the small computer, the word length of the microcommands should be shortened, especially since the word length in this plan has already been limited to 16 bits, thus if condition coding syllables are also set up, there will not be much left. Based on this situation, to solve the selector problem is the key of this plan. Here, a condition signal digit J_{WZ}^{Z} is set up. Its function is described above. Here we must give a follow up explanation of the problem of conditions. Under this situation, the many conditions must be limited: The problem of whether that condition is satisfied or not will only come about when in use. When not in use, all conditions are in the returned to "0" state, i.e., not satisfying any condition. This means when they are not called, they must not be satisfied. Therefore "calls" can be eliminated. When the conditions are concretely analyzed, all of them fall into only two categories: one has only one beat and the other has many beats. For example, transfers due to switching such as K_{XH} and PL_1. For these very few conditions, a coding for "access x condition" must be given by the operational syllable. In this way, these conditions only have one beat.

(3) Determination of the Number of Directions of Transfers

In the design of the operational control microprogram of this machine, there can be five directions for selection. There are four to the selector and one in the

cyclic direction towards maintaining J_{WY}. Here, a retention logic is not set up and the cyclic direction is included in other transfer directions. In this way, there are four selectable transfer directions, i.e., one of the outputs of the four parts on the entry of J_{WZ}. It can be seen from a comprehensive analysis of the microprogram operations table of diagnostic commands that the possibility of having three transfer directions is extremely rare, mostly two transfer directions. Thus the method of inserting an empty box in the microprogram and using a controllable decoding program are both regarded as two directions. In this way, the selector logic is greatly simplified.

For the control selector to be simple numbering of the two directional parts should be limited to 10, 00 of 11, 01. Filling in 10 or 11 in the syllable of J^{β}_{WZ} means the part number of the direction of a successful transfer. The part number of the direction of an unsuccessful transfer is 00 or 01. It can thus be seen that the state of B_{XT} does not change. When a transfer is successful only the state of A_{XT} is changed.



Selector logic diagram (the selector switch is in the dotted block)

Key: (1) Transfer condition

4. Operational Entry Channel Control

Because the limitation on the word length of J_{WZ}, a "lower address" syllable is not set up but is represented by the operational code. The operational code serves as the entry address and as the lower address, i.e., the operational code channel is maintained until that command has completed its operation and then it changes to the common fixed entry 2 for commands to gain access. In general, one command can use up the microcommands of a maximum of four parts. This control is accomplished by J^{β}_{WZ} as previously described simply by writing a "0" for the J^{β}_{WZ} digit of the last operational microcommand of each operational command. In the operational microprogram of some commands, a decision whether to transfer to the fixed entry 2 must be based on the condition (see non- "0" entry command in the Appendix). In this way, J^{R}_{WZ} has two possibilities. For

uniformity, the command must be led into an empty operational microcommand block. If four parts have already filled up the blocks and there are no empty blocks to be used, a logical method can be used as a solution such as the analytical command (see Appendix 1).

After the microprogram in the fixed entry 2 has been executed and completed, a switch to the operational program of that command must be made. This means opening the operational code channel. At this time, "1" is written for the J^{R}_{WZ} digit. But if it is an unconditional transfer command, the action involves computing the transfer address and then sending out a request to access commands. These functions belong to the fixed entry 2, but J^{R}_{WZ} in the microcommand for computing the common address to access data remains constant as "1", and therefore the command cannot be transferred to the common microcommand to access commands, and thus an empty block must be inserted.

The above described the main part of the diagnostic microprogram controller. Some reliability measures must be taken to assure operational reliability, such as strict screening of read only memory components as well as measures of automatic testing.

(III) Automatic Testing of the Hardware Core Units

For the diagnostic program to function properly, the diagnostic components must be automatically tested, called an autotest. An autotest consists of two stages, a longitudinal test stage and a microprogram autotest stage.

1. Longitudinal Test Stage

Testing the read only memory of the diagnostic control memory, the microcommand register and the decoder is called a longitudinal test. It is an accumulation of each digit of all the unit messages in the memory, i.e., obtaining an accumulated sum. To be able to judge the accuracy of the accumulated sum sought, it needs to be compared to a standard result. This standard result consists of all odd parity digits of all unit messages, i.e., a longitudinal set of odd forms. If the accumulated sum obtained at the end of the longitudinal test is "0" then there is no error. Conversely, the accumulated sum may be wrong, or there may be a certain digit in the read only content that is wrong, or maybe a certain digit in the microcommand register is wrong. Such errors can be discovered only after completion of the longitudinal check. Therefore which unit is wrong cannot be known. The duplicate can be used to check the digit where there is an error. The testing of the decoder is conducted similarly by making the decoder reverse its decoding (encoding). The hardware core unit needed for the longitudinal test is the time counter JS_t which can carry out continuous addressing. Computing the accumulated sum is conducted on the modulo 2 plus circuit at the entry point of the microcommand register J_{WZ}. The result is registered in the J_{WZ}. When an error is discovered, the machine is shut off automatically. Then the channel of the modulo 2 plus search is switched off, leaving only the read only direct transmission channel, and using the microcommand coincidence switch, the wrong digit can be checked command by command and syllable by syllable.

2. Automatic Test of the Microprogram

Checking the diagnostic unit and its data channels is called automatic testing of the microprogram. Because the microprogram storage has already been checked and it was found to be correct without any error, its automatic test microprogram in it can be used (it can be called the automatic testing program block) to check other parts. This stage involves the use of the surface board switch data to check 19 data channels and 8 function components divided into 20 test regions. The main data channels to be tested are $J_K \rightarrow J_{ZP} \rightarrow J^Q_{YC}$, $J^Q_{YC} \rightarrow J^Q_{YC}$, the various shift channels of J^Q_{YC}, the intersection channel of J^Q_{YC}, $Q_Z \rightarrow J^Q_{YC}$, $Q_Z \rightarrow JS_Z$, $Q_Z \rightarrow JS_{dd}$, $Q_Z \rightarrow J_{ZP}$, $Q_Z \rightarrow J_{dc}$, $J_{dc} \rightarrow J^Q_{YC}$, $K_7 \rightarrow J_{ZHo}$ etc.

To realize automatic transfer of the two steps of longitudinal test stages and the two stages in the course of automatic testing procedure, two signal triggers C_{JZ} and C_{ZbZ} have been set up. The longitudinal testing stage is indicated by $C_{JZ} = 0$. The automatic test of the microprogram is indicated by $C_{JZ} = 1$.

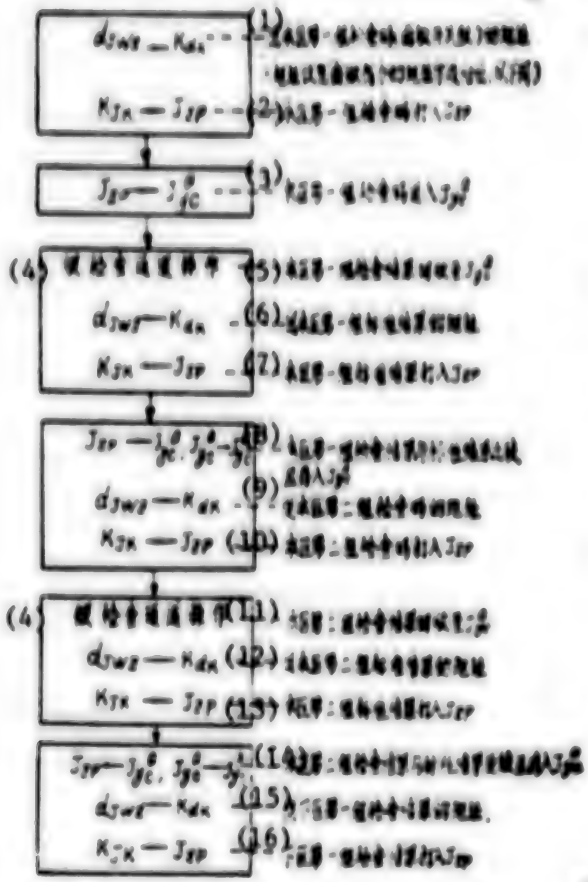
The following is a description of the general structure of the automatic testing of the microprogram. The general form of the channels of automatic testing of the microprogram follows that shown in Flow Chart 1.

Testing the counter is done by taking two as a group each serving as the standard for the other. Its test microprogram is used cyclically. Flow Chart 2 gives the structural form of the automatic checking program of two groups of counters.

By combining the microprograms we can see the thought behind the design: the group of counters JS_{ZHQ} and JS_{sr} with shorter numbers of digits is checked first and then the other group of counters JS_Z and JS_{dd} with longer numbers of digits is checked. After the former is checked, its carry $E_{JS_{ZHQ}}$ and E_{Jssr} are relied upon to make $C^*_{JC} = 1$, and further making $C_{JC} = 1$. In this way, the check transfers to the latter group of counters. After the test, the carry E_{JSZ} or E_{Jsdd} is relied upon to make $C^*_{EJSZ} = 1$ and furthermaking $C_{EJSZ} = 1$. At this point, the cyclic program ends. In addition, the comparison of the status of the counters in the same grou[illegible] conducted in the modulo 2 plus circuit in the J^Q_{YC}, and the comparative [illegible] are stored in J^Q_{YC}. J^Q_{YC} also contains a decide "0" circuit. If the [illegible] "0" circuit is not "0", the machine is shut off, ($Z"0"C_K$), thus indicating [illegible] presence of an error.

After the cyclic microprogram ends, it is transferred to the diagnostic program procedure and clears the status device that has been used to "0". To realize this action of supporting the top and opening the bottom, a fixed entry 1 is set up. It accesses the commands in the surface board J_{K7} to J_{ZP}. As described previously, this kind of commands can serve as a command to shut off the machine or as an unconditional transfer command. It then continues to set the diagnostic process signal trigger C_{gZ}. In this way, the process is automatically converted.

Flow Chart 1



Key:

(1) Address of the first set of bridge checking code (number of surface board switches) sent to this region (The address message is given by the address syllable of the microcommand, same in the following)

(2) The checking code of the first group in this region is entered in J_{ZP}

(3) The checking code of the first group in this region is entered in J^{Q}_{yc}

(4) Channel operation being checked

(5) The results of check of the first group in this region is retrieved to J^{Q}_{yc}

(6) Address of standard result of the first group sent to this region

(7) The standard result of the first group in this region is entered into J_{ZP}

(8) The result of bridge check of the first group in this region is compared to the standard result and stored in J^{Q}_{yc}

(9) Address of checking code of the second group sent to this region

(10) The checking code of the second group in this region is entered into J_{ZP}

(11) The result of check of the second group in this region is retrieved to J^{Q}_{yc}

(12) Address of the standard result of the second group in this region

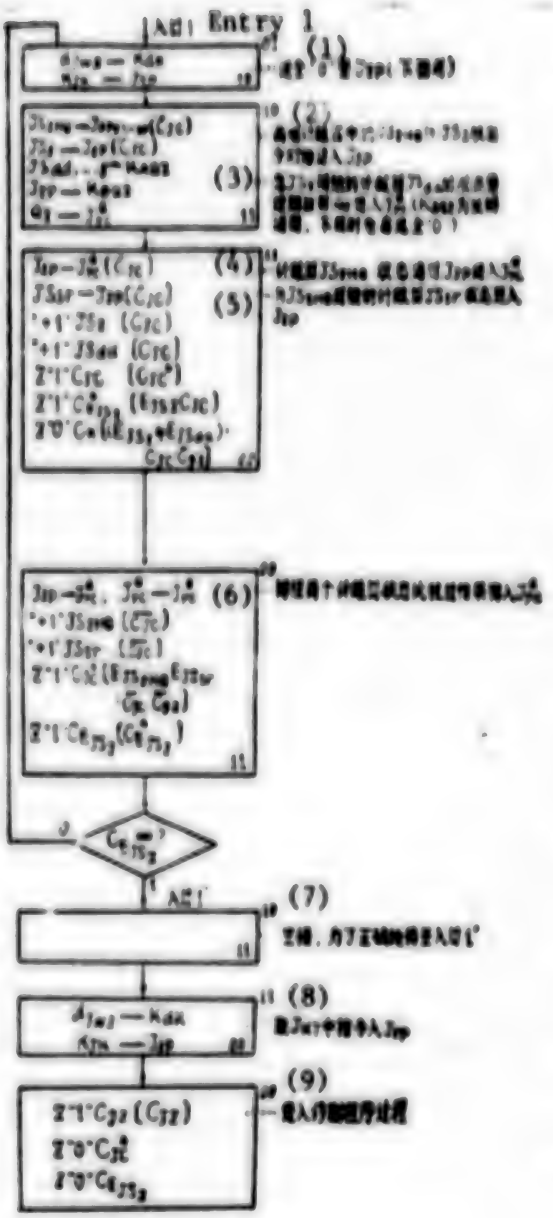
(13) Standard result of the second group in this region is entered into J_{ZP}

(14) Result of check of the second group in this region is compared to the standard result and stored in J^{Q}_{yc}

(15) Address of result of check of the first group sent to the next region

(16) Results of check of the first group of the next region are entered into J_{ZP}

Flow Chart 2



Key:

(1) Send full "0" to J_{ZP} (used below)

(2) The status of JS_{ZdQ} and JS_{Z} in the two sets of address counters are entered into J_{ZP} on a time shared basis.

(3) The status of JS_{dd} address counter of the same set as JS_{Z} is entered into J^{Q}_{yc} via the adder Q_{Z} (K_{BQZ} is the reverse code channel, when not in use, a full "0" must be transmitted)

(4) The status of JS_{ZHQ} address counter enters J^{Q}_{yc} via J_{ZP}

(5) The status of JS_{sr} address counter of the same set as JS_{ZHQ} enters J_{ZP}

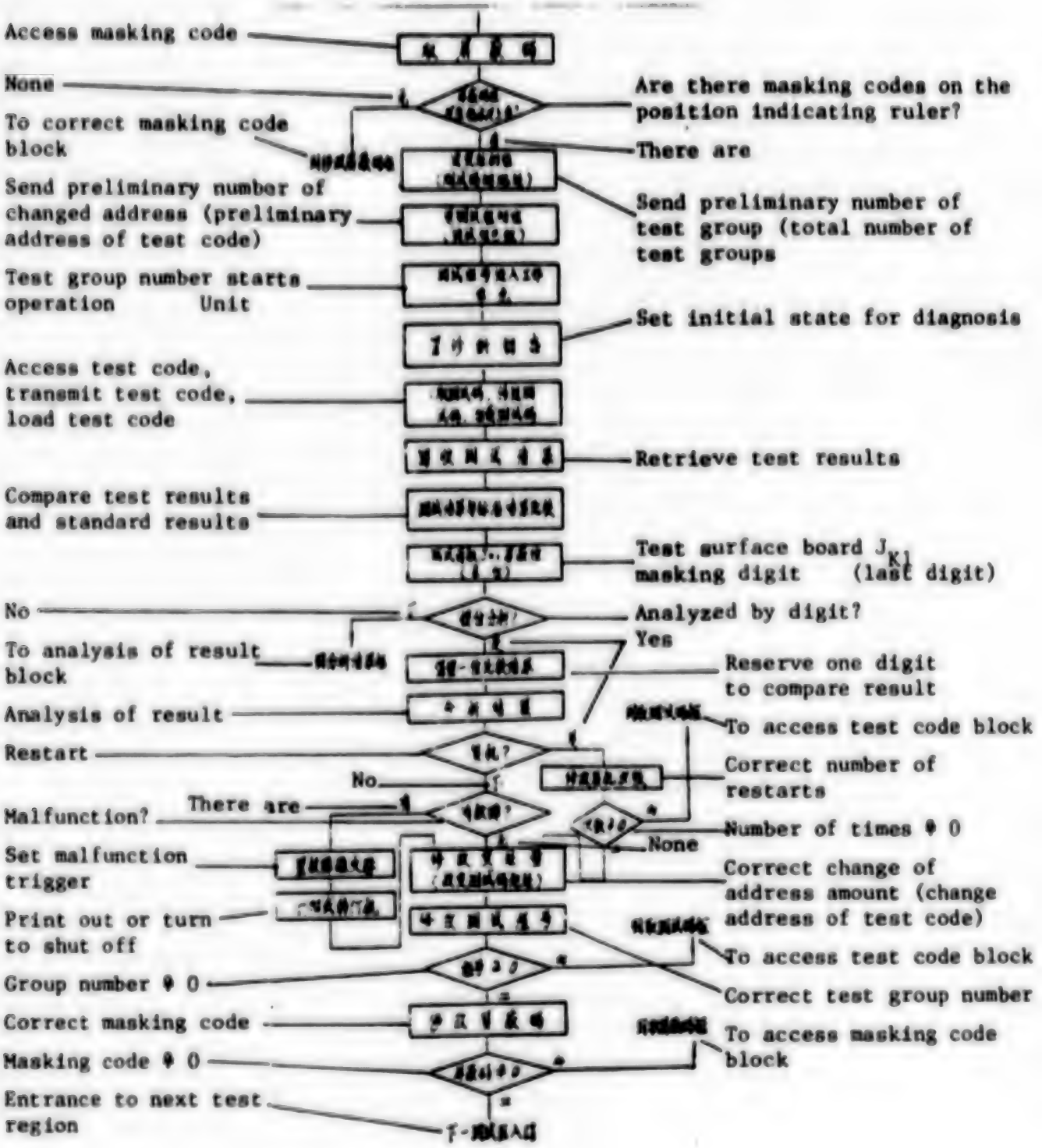
(6) The status of the two address counters of the same set are compared and the results are stored in J^{Q}_{yc}

(7) Empty block for correctly transferring into entry 1'

(8) Access commands from J_{K7} and enter into J_{ZP}

(9) Enter diagnostic program procedure

(IV) Typical Program Block Diagram



V. Software System

(I) Method of Generating the Test Codes

1. General Description

The nature of testing the logic network is to cause a change in the input variable which will uniquely determine the response at the output end. To realize this goal, the channel must be sensitized. Surrounding the problem of sensitization is a series of methods of test code computations, such as the diagrammatic method, Boolean difference method, D algorithm, channel sensitization method, and fault function method etc. The characteristics of all of these methods are different because the methods of computation of sensitization are different, such as whether only a single fault test group can be found but multiple faults cannot be found; whether a hypothetical fault point must first be assumed; whether only the test group of the composite logic can be found but that of the time series logic cannot be found; whether the best test group numbers of single faults can be found etc. The fault function method has solved these problems. But this method is not yet understood by many people at present, therefore it is necessary to briefly introduce it.

2. Variations of the Logic Function

Using the fault function to find the test group is to obtain it by finding variations in the logic function of the tested network. Let the single level logic function under the forbidden character by $Y = Y(x_1, x_2, \ldots\ldots x_n)$, and let its variable state by $\Delta Y = 1 - Y(x_1, x_2, \ldots\ldots x_n)$. How can this variable state extract the test function? Let us look at the following analysis. From the proof we know that $\Delta Y = Y(x_1 + \Delta x_1, x_2 + \Delta x_2, \ldots\ldots x_n + \Delta x_n) - Y(x_1, x_2, \ldots x_n)$.

Thus the variable state of Y is finding the total differentiation of Y. Concretely, the variable state

$$\Delta Y = \sum_{i=1}^{n} d^i Y,$$

This expansion forms the set of functions $[\Delta Y]_Y$ shaped like the structure of Y. The assignment of each $[\Delta Y]_Y$ is an AND term of $\bar{Y}$. Between the terms they are related by OR. Each AND term makes $\bar{Y} = 1$, therefore the set of $[\Delta Y]_Y$, i.e., $\sum_{i=1}^{n} d^i Y = \bar{Y}$. It can thus be seen that the assignment of each $[\Delta Y]_Y$ should all make Y take a false value, thus giving a test of the complementary graph $\bar{G}$ of Y. Therefore $[\Delta Y]_Y$ is called the test function of $\bar{G}$ or $\bar{Y}$. If Y takes a true value, then $\bar{Y}$ contains the fault $[\Delta Y]_Y$ fixed at Y. Thus the expansion of the variable state ΔY of Y is the test function of the fault Y. Similarly, the expansion of the variable state $\Delta\bar{Y}$ of $\bar{Y}$ is the test function of the other type of fault of Y.

It should also be pointed out that ΔY contains the AND terms of i variables of the variable states. Thus Thus the i repeated faults of Y can be detected. When computing the test functions of repeated faults, the original function or th. test function already obtained occurs in the expansion of the variable states, it should be eliminated, because it is not a test function. It can be easily proven that

$$\Delta Y = \bar{Y} = \sum_{i=1}^{n} \Delta^i Y = d^n Y + d^{n-1} Y - Y$$

3. Determination of the Number of Test Functions

(1) The Number of the Best Test Functions

The number of the best test functions of single faults can be obtained. Because the logic network can be described by the logic function, it can also be analyzed graphically and the number of the best test functions can be determined by the graphic method. The number of the best test functions for the logic function of a single level with n variables is $\Lambda + 1 + V + 1 = n + 1$ (where Λ is the number of AND symbols, V is the number of OR symbols, n is the number of input ends used by the components). The number of the best test functions of a multiple level network is $\Sigma\Lambda_i + 1 + \Sigma V_i + 1 = \Sigma(\Lambda_i + V_i + 1) - \Sigma 1 + 2$. The number is the total number of input ends used by the elements plus 1 minus the total number of output ends (Σ 1 represents the total number of output ends).

(2) The Number of Complete Test Functions

Generally, the best test function of single faults is used to diagnose single faults. If multiple faults have to be diagnosed, it will also give a test function. It is a complete test function. Its number is $\bar{Y}(1) + Y(1)$, i.e., adding the values of the two algebraic polynomials obtained by substituting 1 for all the variables in the original function Y and the complementary function $\bar{Y}$ of the single layer logic function. In the channel diagram, it is the total of the number of channels of all the directional branch channels of the original diagram and the complementary diagram and all other directional branch channels connected together once. This is its nature.

The following example is given:

Find the number of diagnostic test functions of

$$Z = x_1 x_2 \ldots x_p + y_1 y_2 \ldots y_q.$$

Solution: The number of input ends used is $n = p + q$. Thus the number of the best diagnostic test functions of single faults is:

$$T'_H = n + 1 = p + q + 1$$

Since $\bar{Z} = (\bar{x}_2 + \bar{x}_2 + ... + \bar{x}_p)\ (\bar{y}_1 + \bar{y}_2 + ... + \bar{y}_q)$, the number of the diagnostic test functions of all the faults is:

$$T_H = \bar{Z}(1) + Z(1) = pq + 2$$

4. Method of Finding the Variable States of the Test Function and Locating the Faults

(a) Method of finding the variable states of the function

We know that the logic function can be expressed in the form of the sum of products and it can also be expressed in the form of the product of sums. Here, as used previously, it is assumed that the foregoing is taken as the standard to facilitate the discussion. The complementary diagram of the logic function in the form of the sum of products has a serial structure. The sequence of test functions formed by the greatest AND terms in the ΔY variable states can connect the greatest parallel subdiagrams in the complementary diagram, thus it can test the entire diagram of the complementary diagram. Therefore this sequence of test functions is the test function. The number of this sequence is $\Lambda_M + 1$ (where Λ_M is the greatest number of AND symbols of the greatest AND terms). Under ordinary conditions, the number of test functions does not equal the number of the best test functions. At this time, the test function can only detect faults but cannot locate. For this, a locating test function must be added. Its number is $(\Lambda + 1) - (\Lambda_M + 1)$. The added locating test functions are for partially changing the channels that have already been tested previously to facilitate locating. The actual method is to use a certain AND term of a certain test function to replace the corresponding but different term in other test functions.

It should also be pointed out that as the variables of the greatest AND terms take on variable states, and the variables of the AND terms which are not the greatest have all taken on variable states and there are no more new variable states to be taken on, the variable state of the variable on the extreme right side is taken. The test function thus connects the branch channels on the extreme right side of the serial subdiagram that is not the greatest more than once, i.e., "borrowing channels." In this way, with a test function, a locating function can be generated.

After the test function has been obtained, a test sequence for the assignment of variables can be obtained. The principles of assignment of variables is as follows:

(1) If the test function does not contain complementary variables of variable states, then variables are assigned according to nonvariable states and variable states.

(2) If the test function contains complementary variables of nonvariable states, then the following assignment principles are added:

(i) If $[dY]_Y$ has variables of complementary states, the assignment must be able to point out the problem and should correspond to the problem channel in the complementary diagram, therefore it is not the test function of $\bar{Y}(Y)$ and should be discarded and taken anew.

(ii) If the complementary variables of $[dY]_Y$ are not all variables of variable states, then variables should be assigned according to the variable states. After assignment, the number of variables of variable states of the AND terms in the variables of complementary nonvariable states increases, providing multiple channels to the corresponding subdiagrams in the complementary diagram and forming potential channels.

(iii) If the complementary variables of $[dY]_Y$ are all variables of nonvariable states, then the complementary variables should be individually assigned to transfer the potential channels and isolate the single faults.

The actual logic circuit frequently has multiple levels. Logic networks of multiple levels are represented by multiple order logic functions while high order logic functions are expressed by complex functions. Therefore, differentiation of the complex functions can be used to perform step by step replacement of variables of the functions from output to input and to solve the variable states, thus forming the test functions of each level.

(b) Locating the fault

When the test function $[\Delta Y]_Y$ shaped like the Y structure on the same level is compared to the original function, the state of the variables in the original function is the point of fault. Therefore, the fault function is obtained and is called $[\Delta Y]^*_Y$.

Summarizing the above, fault function diagnostic techniques belong to fault diagnosis. It uses the graphic method to determine the number of the smallest test function and it uses the algebraic method to determine the actual test function and it uses the logical method to determine the faulty part.

(c) An example of application

Diagnose the network described by $Y = x_1x_2x_3 + x_4x_5$

Solution:

$$\overline{Y} = (\bar{x}_1 + \bar{x}_2 + \bar{x}_3)(\bar{x}_4 + \bar{x}_5)$$

$$dY = dx_1x_2x_3 + dx_4x_5 \cdots\cdots ①$$

$$x_1dx_2x_3 + x_1dx_3 \cdots\cdots ②$$

$$x_1x_2dx_3 + x_4dx_5 \cdots\cdots ③$$

$$x_1dx_2x_3 + dx_4x_5 \cdots\cdots ④$$

$$d\overline{Y} = d(\bar{x}_1 + \bar{x}_2 + \bar{x}_3)(\bar{x}_4 + \bar{x}_5) \cdots\cdots ⑤$$

$$(\bar{x}_1 + \bar{x}_2 + \bar{x}_3)d(\bar{x}_4 + \bar{x}_5) \cdots\cdots ⑥$$

Equation (4) is obtained by replacing x_4dx_5 of equation (2) by dx_4x_5 of equation (1).

The test and fault functions are shown in the following table:

Test

$(dY)_Y$ / $(d\bar{Y})_Y^-$	x_1	x_2	x_3	x_4	x_5	Y	$(\Delta Y)^*_Y$ / $(\Delta\bar{Y})^*_{\bar{Y}}$
①	0	1	1	0	1	0	$Y^1 = x^1_1 + x^1_4$
②	1	0	1	1	0	0	$Y^1 = x^1_2 + x^1_5$
③	1	1	0	1	0	0	$Y^1 = x^1_3 + x^1_5$
④	1	0	1	0	1	0	$Y = x^1_3 + x^1_4$
⑤	1	1	1	0	0	1	$Y^0 = x^0_1 + x^0_2$
⑥	0	0	0	1	1	1	$Y_0 = x^0_4 + x^0_5$

(II) Special Diagnostic Commands

This machine has 25 diagnostic commands shown in detail in Appendix 2. Here, the meaning of only a few special diagnostic commands is explained.

1. Installed Microcommand W_1

(1) FORM

CM_8	$Empty_3$	B_1	d_{20}

(2) MEANING

1) $(D) \Longrightarrow J_{wy}$,65~96
2) $(D+4) \Longrightarrow J_{wy}$,33~64
3) $(D+4) \Longrightarrow J_{wy}$,1~32

(3) USE To prepare the test code the operational control microcommand register J_{wy} has to be equipped with an encoder for microaction signals and only one beat is used. It is used in combination with the loaded test code command CS.

2. Loaded Test Code CS

(1) FORM

CM_8	CB_3	B_1	d_{20}

(2) MEANING

1) $(D) \Longrightarrow J_{zp}$
2) $CB_i \Longrightarrow C_{ci}$

3) Z"0"C_Z (the diagnostic function clock is turned off and the operational control function clock is turned on).
4) Z"0"C_{ZY} turns on the output of the operational control decoder.

(3) USE To access the test code for the part to be tested and to place it in J_{ZP}, its previous command is W_1. In this way, the test code is loaded onto the tested parts with a length of 32 digits or 64 digits.

3. Installed Microcommand W_2

(1) FORM

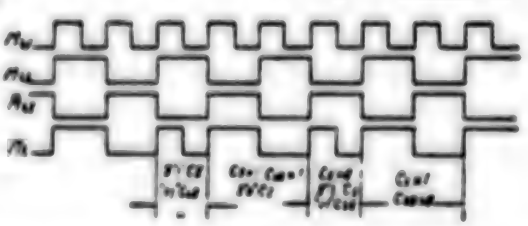
CM_8	CB_3	B_1	d_{20}

(2) MEANING

1) $(D) \longrightarrow J_{wy}65\sim96$
2) $(D+4) \longrightarrow J_{wy}33\sim64$
3) $(D+8) \longrightarrow J_{wy}1\sim32$
4) $CB_1 \longrightarrow C_{z1}$ $CB_2 \longrightarrow C_{z2}$
5) Z"0"C_Z (turns off the diagnostic function clock and turns on the operational control function clock)
6) Z"0"C_{ZY} (turns on the output of the operational control decoder).

(3) USE To let the test code go through the old test region or to retrieve the test results to J^{Q}_{yc}, the J_{wy} requires an installation of an encoder for signals of microaction, and one beat or two beats can be used (decided by CB_2).

When executing the CS and W_2 commands, there is a problem of shutting off the diagnostic function clock and turning on the operational control function clock. This is because the diagnostic function cycle is two times the normal cycle of operational control functions (avoiding a shortage of the diagnostic logic levels), but during the test, the original clock cycle of operational control must be used (to truly test the fault caused by a shortage of the logic level). Here a problem concerning transformation of the clock cycles exists. Also, a problem concerning the selection of the polarity of the clock exists. These are explained in the following diagram.



Selection of polarity of the wave forms

The relative position of the wave forms shows that the following logical relationship is satisfied:

$$\bar{M}_t = M_{ty}\bar{C}_Z + M_{tZ}C_Z\bar{C}_{bZ} + \bar{M}_{tZ}C_ZC_{bZ}$$

Of these, $C_Z = 0$ indicates a state of operational control function; $C_Z = 1$ indicates a state of diagnostic function; M_{ty} is the operational control function clock; M_{tZ} is the diagnostic function clock; C_{bZ} is the control trigger for the selection of wave forms, M_t is the combined function clock.

4. Test Results of Retrieval

(1) FORM

CM_8	$Empty_3$	B_1	d_{20}

(2) MEANING $(D_{15}\cdots\cdots D_0) \longrightarrow J^Q_{yc}$

(3) USE The retrieval points are the tested units that are unrelated to the bus switch, such as J^H_{yc}, $J_{WY}1 - 32$, $J_{WY}33 - 64$, J_{WY} 65 - 96, YH_{WY} and spare parts.

5. Analysis of the results FX

(1) FORM

CM_8	$Empty_3$	B_1	d_{20}

(2) MEANING
1) If $(J^Q_{yc}) \neq 0$, Z"1"C_{yg} and $K_{CT} = 1$, the machine is turned off;
2) If $K_{XH} = 1$, then turn to D;
3) If $K_{XH} = 0$ and $(J^Q_{yc}) \neq 0$ and $K_{CT} = 0$, then execution is in order;
4) If $K_{XH} = 0$ and $(J^Q_{yc}) = 0$, then jump one access command.

(3) USE Used for analyzing the results, the decide "0" results of J^Q_{yc} and the states K_{CT} and K_{XH} are combined, giving four directions of operation.

(III) Mode of Programmed Operation

Command access and data access are serial operational modes. The diagnostic program transfers the program from the data tape to the internal memory by the input command. Then it is placed at program entry and execution of the program begins. The commands go from the internal memory to J_{ZP} via the K_7 switch of the memory control. Data also go to J_{ZP} via the K_7 switch and then are stored in J^Q_{yc}. Operation then begins. This is the mode of operation. Input and execution of the program are carried out entirely under the control of the diagnostic microprogram.

VI. Function of the Diagnostic System

A complete diagnostic machine must have three functions: first used as a debugging tool, then used as a monitoring tool during testing and afterwards used as a maintenance tool. This system does not possess the second function. Only the special diagnostic machine in large computers has such a function. They can discover errors in time via a complete checking system and report the error to the diagnostic machine. The diagnostic machine can start the diagnostic subroutine of the corresponding part based on the code of the checking signal (like interrupt registration) to carry out diagnosis. If there are support units, the faulty units can first be replaced and then diagnosed off line. If there are no support units, the field must be protected first and then diagnosis can be performed. After the malfunction has been eliminated, the field is restored.

This system only possesses the first and third functions.

(I) Debugging Function

Because the hardware core unit is a part of the command control unit, it can only perform the function of debugging the operational control unit and debugging command control unit. After one part in the main memory is debugged and under the prerequisite that the joint debugging of the memory control and the main memory has been completed, the diagnostic program can be used to debug the operational control unit. This is exactly the same as diagnosis during the maintenance stage. Technical faults can be detected such as broken (missing) lines, multiple (wrong) lines, welding of wrong places and missing places that should have been welded. Malfunctioning of components and faults in logic can also be detected. If location test codes are more finely tuned, accurate locating is possible.

Inserted units can be used for debugging and can also be used to debug this machine in a snowball fashion. The inserted unit is inserted into one's own position and the diagnostic program is then connected to it. This is all that is required.

Debugging is done in two steps because the method of computing the test codes are different. The first step is the fault debugging stage. The test code is theoretically obtained according to the method of computation. The second step is the stage of checking the function. At this time, the test codes are the same as the function checking codes. They are used to check errors in the method of computation, accuracy, command functions and logic design. The efficiency of locating in this step is not as good as the previous step and human intervention is needed.

According to this debugging order, the operational control unit does not need to have separate debugging equipment. If there is an independent diagnostic machine the debugging function will not be limited by the order of debugging.

(II) Maintenance Function

When the machine malfunctions, it is shut off. When the machine is turned on and the system is diagnosed, the malfunction can be quickly located on 1 to 2 inserted units. Supporting inserted units can replace the faulty ones and the machine can continue to operate. Thus, a highly efficient maintenance tool is an important means to increase the machine's usability rate and reliability.

VII. Function and Characteristics of the Diagnostic System

(I) Function

1. Area Covered

The percentage of diagnosis of operational control is 100 percent in principle. But actually because of the lack of connection between some spare units and the bus switch, or because of too many turn abouts, it is not convenient to sweep in or retrieve, the actual diagnostic percentage is over 95 percent. The diagnostic percentage of the main frame is about 80 percent, because the memory control has its own means of automatic testing and the memory control and main memory have a means for joint debugging. The main function units of the command control unit can serve as the hardware core for diagnosis and thus undergo automatic testing. The important accumulator register of the command control unit can also be diagnosed. The operational control unit is almost entirely diagnosed and these occupy most of the equipment of the mainframe.

2. Locating Efficiency

It is closely related to the method of computation of the test codes. The efficiency of locating the fault using the "fault function method" to locate the fault locating code is very high, because it is actually a microchannel diagnosis. Therefore it can locate the fault fixed at "0" or at "1" in each component. The use of function checking codes to locate the faults is in principle not as effective as the former but the efficiency of locating faults of this type of checking codes is closely related to the structure of the machine. If the structure is rational, a very good locating efficiency can be achieved. If a function code is used to test the read only memory of the operational control unit, then the fault locating can be accurate to within a medium scale component.

For this machine, when necessary, diagnosis can locate a fault to within an inserted unit, in very few cases the faults can be located within two inserted units. Function test locating can locate faults within a component.

3. The Number of Test Regions

This machine has about 50 test regions.

4. Efficiency of Test Codes

The number of test groups for faults detected and located by the "fault function method" is the best and they are not repetitive. It can detect and locate single faults and can also detect and locate multiple faults. Therefore that test code has a higher efficiency. Function test codes also have a better locating efficiency under definite conditions but because more test groups are needed, the test efficiency is not as good as the former.

5. Types of Detected Faults

This machine can detect fixed faults and can also detect intermittent faults. This is realized mainly by repeated testing. A special row of switches is set up on the surface board. Eight of the digits are used to establish the number of repeated tests. In addition, errors due to malfunction can be detected and functional errors can also be detected. Because the use of diagnostic commands can debug or can be used to install the operational control microprograms of any part, this is the same as diagnosing ROM_Y.

6. Truthfulness of Diagnosis

Because a diagnostic retrieval switch has been set up, the logic levels have been increased, thus the diagnostic cycle has increased to two times the operational control function cycle. Diagnostic truthfulness can only be possible in this way. In this way, the faults caused by a shortage of the logic levels of operational control can be discovered. For this, a function cycle transfer and selection and control of the polarity of wave forms are set up.

7. Amount of Diagnostic Equipment

Because the command control function unit is shared, the number of special diagnostic units is reduced. Only 20 inserted units (mostly standard units, there are only two nonstandardized diagnostic units). The operational control unit has 200 inserted units (the inserted units of the accumulator register being diagnosed are not included, they belong to the command control unit). Thus the amount of diagnostic equipment constitutes 10 percent. If there are more nonstandardized inserted units for more types of diagnosis and when diagnosis of parts of nonoperational control units are counted, the percentage is even lower. This is already within the limits of the international statistical standard of 25 percent.

(II) Characteristics

1. Diagnosis of this system is microdiagnosis. It is a diagnosis of microchannels through the implementation of microcommands.

2. This diagnostic system is actually a hidden diagnostic machine. Because it uses the command control unit as the hardware core, and it also uses memory control and main control, therefore it is not an independent machine, but there is a set of software systems and hardware systems (even some are used in combination) that possesses all diagnostic functions, thus it is a small diagnostic machine contained within the mainframe.

3. The software has a diagnostic command system. It is more versatile than the diagnostic program block.

4. Automatic testing is automatically converted to diagnosis and diagnosis must first undergo autotesting. This is a measure to assure reliability and dependability.

. Transfer of diagnostic program can be carried out in a versatile manner by switches. The row of switches on the surface board has 24 digits for such use.

6. The number of test regions and the number of groups of test regions are comprehensively considered, because the number of test regions is not only related to the number of logic function blocks but also is related to the number of test groups. If the test region is too large, the number of test groups increases. If the test region is too small, the number of test groups is reduced, and the number of test regions is increased, therefore they must be uniformly considered to achieve a faster diagnostic speed.

7. Special consideration has been given to shorten the word length of the microcommands and the correct selection of the parts.

8. The diagnostic system and the system being diagnosed are well coordinated. Here, microdiagnosis is matched with the operational control unit's microprogram so that the microaction signals for operational control needed in the test are easily accessed. The operational control unit has an uncontrolled microcommand decoder, otherwise the diagnosis must drop the various control conditions on side channels, the microcommand register has specially added some microaction signals for diagnosis to facilitate retrieval or testing, such as the addition of the signal of the test accumulator register J_L; the main register operational control and the bus are connected by a double channel structure (far and near) to facilitate retrieval.

9. The design of the diagnostic system takes into consideration the entire situation of the processor and combines hardware and software. For example, the "fault function method" regards the clock M_t as a variable when diagnosing the time sequence logic. Thus a logic to control the clock M_t is added to the hardware to assure normal operational time is counted by M_t and a test code can be added for the time of diagnosis. Diagnosis of spare parts uses hardware with a retrieval switch and added software combined with the use of masking techniques. Because spare parts are not directly connected to the bus, a retrieval channel is added to retrieve those parts that are truly not easy to retrieve. All retrievable parts are all retrieved and compared using the locating indicator ruler's masking function. Using the function test code to test the operational control unit's microprogram storage has achieved a better locating effect. This is also an actual example of combining software and hardware.

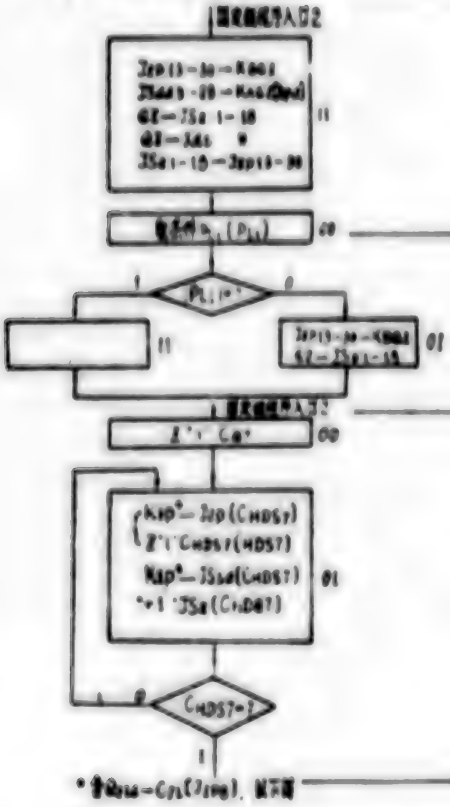
VIII. Concluding Remark

The establishment of this diagnostic system for the diagnosis of digital computers is a preliminary attempt.

Because the hardware core shares the command control unit, the net increase of equipment is very small, the cost is low, and practice proves that the system has a practical value in the diagnosis of medium and small computers.

Appendix I

1. 非 "0" 转移：——Non- "0" transfer 4：



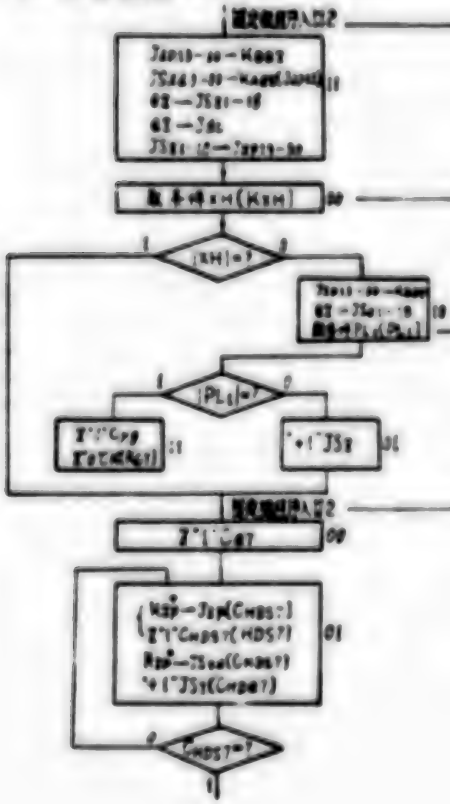
Fixed microprogram entry 2

Access condition P_{L1} (P_{L1})

Fixed microprogram entry 2

Contain Q_{Z14} C_{JL}(J_{ZP8})

2. 分析结果 FX，——Result of Analysis FX:



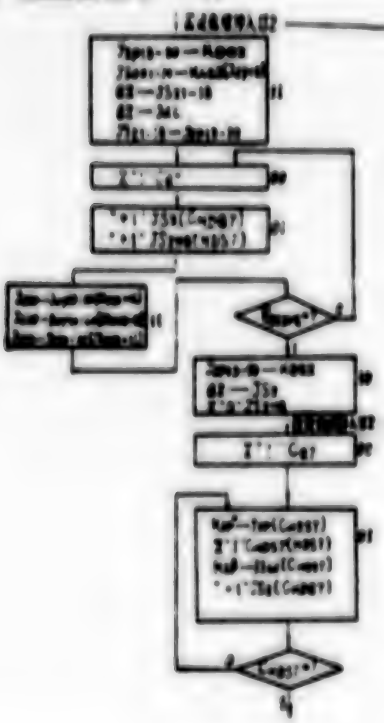
Fixed microprogram entry 2

Access condition XH (KXH)

Access condition PL_1 (PL_1)

Fixed microprogram entry 2

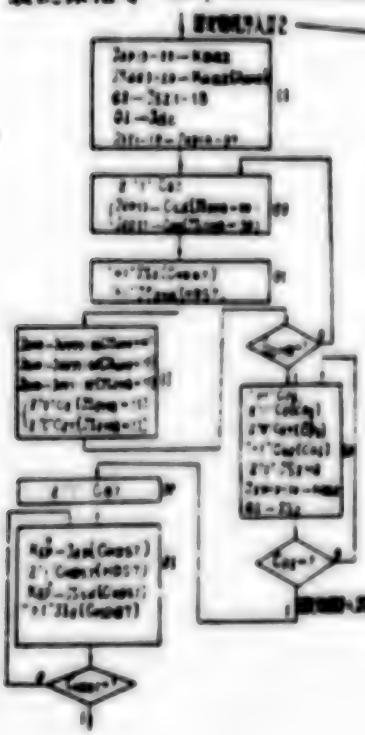
3. 装配微指令 W_1 —— Installed Microcommand W_1:



Fixed microprogram entry 2

Fixed microprogram entry 2

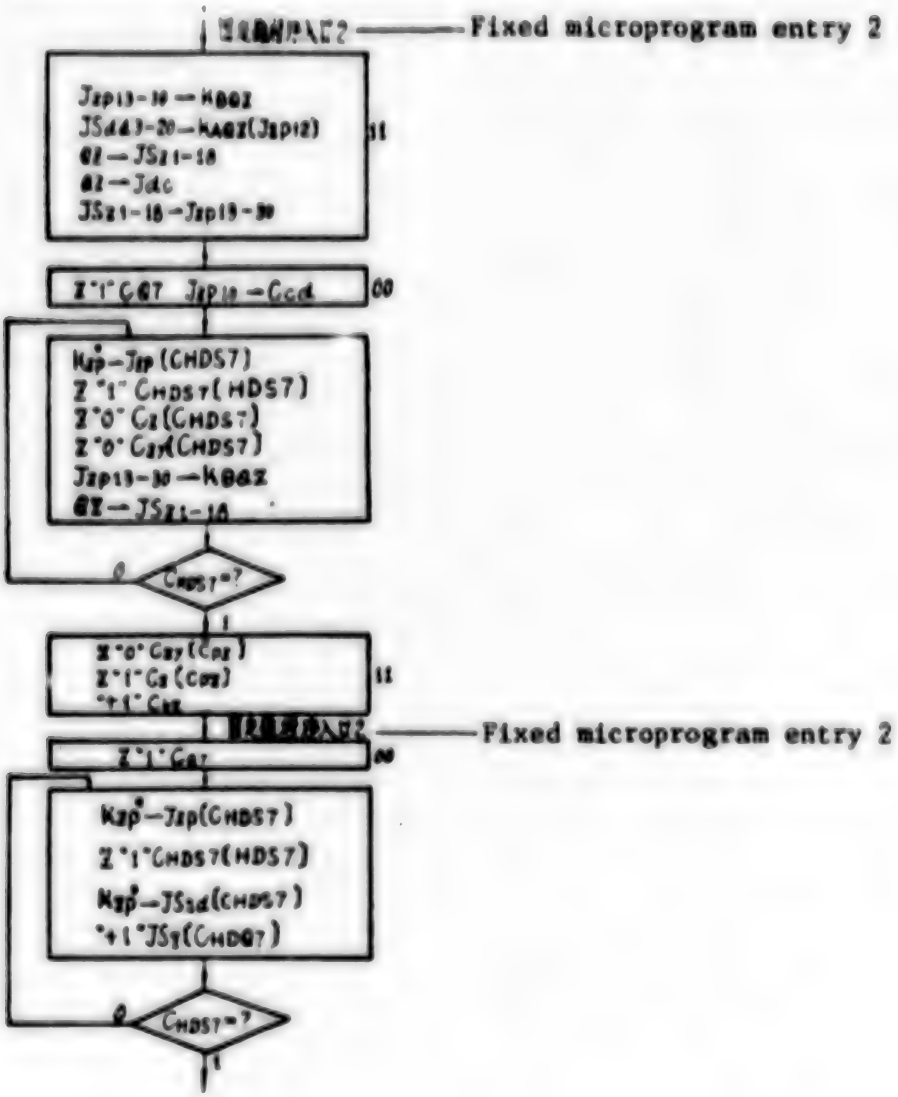
4. 装配微指令 W_2 —— Installed Microcommand W_2:



Fixed microprogram entry 2

Fixed microprogram entry 2

5. 加载测试码 CS₁ ——————Load Test Code CS:



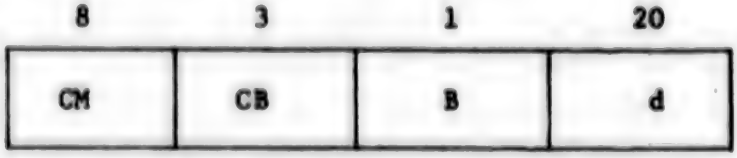
Appendix II Diagnostic Commands System

1. Type:

The diagnostic commands system can be divided into the following types according to the modes of operation:

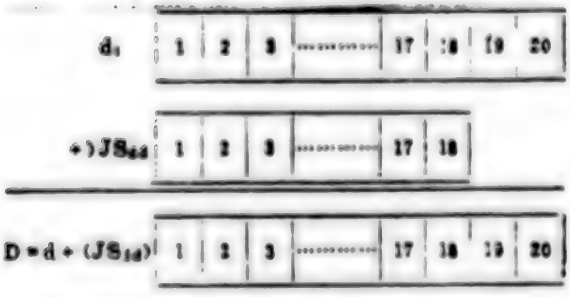
(1) Memory reference
(2) Test of general purpose adder (J_L)
(3) Retrieval
(4) Shift
(5) Other

2. Command Format:



Where CM--operational code of 8 digits
d--relative address 20 digits
B--address change indicator digit 1 digit

$$\text{Reference address } D = \begin{cases} d & [B = 0] \\ d + (JS_{dd}) & [B = 1] \end{cases}$$



i.e.: The high 18 digits and the 18 digits of JS_{dd} are lined up and added to obtain high 18 digits, the last 2 digits of D, D_{19} D_{20} are the same as d_{19} d_{20}.

D is written as D_1 D_2 D_3.....D_{17} D_{18} D_{19} D_{20}

In memory reference type commands, the high 18 digits of D are D_1 D_2 .....D_{18}, representing the address in the main memory, the remaining are not defined.

In the test J_L type command, the D_{14} D_{15} D_{16} D_{17} D_{18} of D represent different channels and different J_L, the rest are not defined.

In the retrieval type command, D_{15} .....D_{18} of D represent the coded address of the retrieved parts (test region), the remaining are not defined.

In shift type commands, D_{17} D_{18} of D respectively represent the direction of shift and the number of shifts, the remaining are not defined.

In other types of commands, D is not defined.

--CB--operational code indicator digit, a total of three digits, separately called CB_1 CB_2 CB_3 , which are only meaningful in W_2, CS, + CB.
--OB has the following meaning in W_2 and CS commands:
--CB_1--long and short operation indicator digit;
--CB_2--single/double beat operation indicator digit;
--CB_3--side route indicator digit.

In the +CB commands, its meaning represents constants 0-7.

3. Operational Modes of the Commands:

The commands are of single address forms, under general conditions, the first operational number is in J^Q_{YC}, thus, the basic form of operation of the commands is:

$$(J^Q_{YC})CM(D) \Rightarrow J^Q_{YC}$$

4. Form and meaning of each type of commands:

1. Memory reference type

(1) Access the operational number from the main memory ↓

Form:

8	3	1	20
↓	empty	B	d

Meaning: $(D) \Rightarrow J^Q_{YC}$

(2) Input numbers into main memory ↑

Form:

8	e	1	20
↑	empty	B	d

Meaning: $(J^Q_{YC}) \Rightarrow D$

(3) Logic multiplication

(1) 形式：

8	3	1	20
∧	3空	B	d

(2) 意义：$(J^{Q}_{YC})\wedge(D)\Rightarrow J^{Q}_{YC}$

(4) ⊕ 模二加

(1) 形式：

8	3	1	20
⊕	3空	B	d

(2) 意义：$(J^{Q}_{YC})\oplus(D)\Rightarrow J^{Q}_{YC}$

(5) 无条件转移 ↱

(1) 形式：

8	3	1	20
↱	3空	B	d

(2) 意义：(6)

(6) 转向D取指令

(7) (8) 转子

(1) 形式：

8	3	1	20
转子 8	3空	B	d

(2) 意义：(9)

(9) 下一条指令地址 $(JS_{Z})+4\Rightarrow D$

(10) 转向D+4 取指令

(11) 非零转移 ≠

(1) 形式：

8	3	1	20
≠	3空	B	d

(2) 意义：(12)

(12) 当 $(J^{Q}_{YC})\neq 0$ 时转向D取指令

(13) 当 $(J^{Q}_{YC})=0$ 时，顺序取指令

(8) 传送B，B↓

(14)

(1) 形式：

8	3	1	20
B↓	3空	B	d

(2) 意义：(D) 左端18位 (D) 1,2……18⇒JS_{dd} (15)

(15) The first 18 digits of

(16) 测试 B1，≠B1

(1) 形式：

8	3	1	20
≠B1	3空	B	d

(2) 意义：①$(JS_{dd})+1\Rightarrow JS_{dd}$

(17) 当 $(JS_{dd})\neq 0$ 时转向D取指令

(18) 当 $(JS_{dd})=0$ 时顺序取指令

(16) Test B1, ≠B1

(17) When $(JS_{dd})\neq 0$, transfer to D to access commands

(18) When $(JS_{dd})=0$, access commands in order

(19) 测试 B2，≠B2

(1) 形式：

8	3	1	20
≠B2	3空	B	d

(2) 意义：①$(JS_{dd})+(D)\Rightarrow JS_{dd}$

(20) 当 $(JS_{dd})\neq 0$ 时顺序取指令

(21) 当 $(JS_{dd})=0$ 时跳一条取指令

(22) (1) 保存B，B↑

(19) Test B2, ≠B2

(20) When $(JS_{dd})\neq 0$, access commands in order

(21) When $(JS_{dd})=0$, jump one to access commands

(22) Retain B, B↑

(1) 形式

8	3	1	20
B↑	3空	B	d

(2) 意义：$(JS_{dd})\Rightarrow(D)$ 1,2……18其余位清零 (23)

(23) 18 the remaining cleared to zero

(24) (12) 加D指令 +D

(1) 形式

8	3	1	20
+D	3空	B	d

(24) Add D command +D

Key: (1) Form:

(2) Meaning: (3) digits

(4) Carry delete

(5) Add CB command +CB

(6) the remaining cleared to zero

(7) Installed Microcommand W_1

(8) Installed Microcommand W_2

(9) ① is same as W_1

(10) is long and short command indicator trigger)

(11) is single/double beat indicator trigger)

(12) is side route indicator trigger)

(13) (close diagnostic work cycle, open up operational control work cycle)

(14) Load test code CS

(15) ② Same as W_2 ②

(16) ③ Same as W_2 ③

(17) Result of Analysis FX

(18) Empty

(19) ① if K cycle = 1, then transfer to D to access commands

(20) ② if K cycle = 0, when $(J^{Q}_{YC}) \neq 0$ and K malfunctions = 1 hour,

(22) Z"1"C_{YS} and machine is shut off

(23) When $(J^{Q}_{YC}) \neq 0$ and K malfunctions = 0, access commands in order

(24) When $(J^{Q}_{YC}) = 0$, jump one to access commands

(25) Retrieval command HS

(26) Shift command $\rightleftharpoons$

Key:

(1) Form:

(2) Meaning:

(3) General purpose adder J_L for test

(4) Empty

(5) content exits/enters via Li channels

(6) content exits/enters via Lj channels

(7) respectively represent the addresses of 16 different adders

(8) Shut off command TJ

(9) Photoelectric input SR

(10) Narrow line printer output

(11) Set initial value

(12) Return to autotest initial state

9296
CSO: 8111/0224

303-1 GENERAL WAGE PROGRAM INTRODUCED

Beijing DIANZI KEXUE JISHU [ELECTRONICS SCIENCE AND TECHNOLOGY] in Chinese No 4, 4 Oct 80 pp 15-16

[Article by Chen Shengdong [7115 5116 2639]: "An Application Software of the DJS100 Series Computer; Brief Introduction to the Function of the '303-1 General Wage Program'"]

[Text] Using the computer to compute wages has already been practiced in our nation. The ordinary computational programs are mostly written in ALGOL, COBOL and BASIC, which are high level languages. Using high level languages to write programs has its advantages but it also brings about some problems. For example, data input is mostly in the form of source tapes and during the input process, mistakes can easily be made, and the procedures of operation are also more complex. More importantly, programs written in high level languages frequently can only be used on large machines, or require larger internal storage, therefore are not easily popularized on small machines. According to the present situation in our nation, it is necessary to design more applications programs suitable for small machines, and to expand the scope of application.

The "303-1 general wage program" is a wage computation program for use on small computers. The program is written in assembly language. When used in the DJS-130 machine, it takes up 2K of internal memory, and including data regions and operational elements, any DJS1--series machine with over 4K of internal memory--can use that program.

In the process of designing the program, the emphasis has always been on "general purpose," therefore that program is highly versatile and general. During the design, the 14 output formats were designed in reference to the wage forms used domestically and the actual situation of the output equipment of the 100 series machines. These include: H -- work number (work card number or code number); G -- basic wage; J -- added wage (or activity wage); X -- cleaning costs (or social benefits costs); E -- transportation costs (or other income); Z -- total income; F -- room fee; B -- nursery fee (child care fee); C -- meal tickets; N -- heating fee; Y -- medical costs; K -- deductions (sick and business leave); Q -- other expenses; S -- net salary.

The letters listed above are the first letters of the pinyin of Chinese characters of the various wage parameters. They not only serve as the names of the parameters but also the commands of input of those parameters.

Actually, each wage parameter constitutes one row in the parametric matrix. The program requires the user to assign a numerical order to each item to be computed as the pointer of the line of the parametric matrix (or called internal codes), and the command A is used to represent it. The program realizes the input of any wage parameter of all items to be computed. For example, a person Jia is assigned as internal code 012, his basic wage is 62 yuan, his activity wage is 6.5 yuan, medical cost is 1.5 yuan, telephone cost is 2.3 yuan, then, the input string of commands is:

A12*G6200*J650*Y150*0230

Here, the parametric values are based on the monetary unit "fen" of the Renminbi, "*" is the sign to separate the numbers. Input of the string of commands can be teletype input and photoelectric input.

Consideration has been given to the fact that the wage forms of each unit contain some relatively fixed items and several variable items. To improve the efficiency of the computer, the program requires the user to record the fixed items on paper tape (also called the fixed tape) for the computer. The number of fixed items can be decided by the user. The program requires the user to dial in the exact number of items into the 01 unit. The fixed tape can be of two tapes--tape A and tape B.

The type A fixed tape is a group of numbers headed by the command R. Each number in each group of numbers represents one wage parameter and the numbers are separated by the separation symbol "*". The program processes the command R taking N items as a group (i.e., content of the 01 unit -- number of fixed items), and storing the data in order into the parametric matrix. Thus, for a user with N = 8,,

R701*5500*300*160*0*0*350*356*702*6200*300*160*250*0*456*124*

equivalent to:

A0*H701*G5500*J300*X160*E0*F350*B356

A1*H702*G6200*J300*X160*E250*F456*B124*

Here it must be pointed out that:1) the total income (Z) is computed by the program from the previous items of income. At the time of input, it serves only as a format parameter, and it can be replaced by any numerical value, but like other zero value terms, it must occupy the space of one term; and 2) R commands are stored in order beginning with the number zero in the order numericals. Thus on the fixed tape, it is not necessary for any other commands to appear. The A type paper tape, whether it is a fixed tape or a variable tape (a paper tape that includes strings of commands of variable items), can use both the 5-unit and the 8-unit formats.

The program provides the function of manual inspection of mistakes for the A type paper tape input. The program can affect the demands of the user by making all messages contained on the paper tape to reappear.

The B type fixed tape is a special binary purpose format paper tape punched out according to the keypunch commands (ZB) of this program. Input of binary fixed tape is executed by a special command (W). At this time, the machine itself inspects the mistakes.

Because of the different management systems of each department at present, the "303-1 general purpose wage program" has selected a more general purpose management form to realize the wage computations of a three level management system. The program takes the shift group (a maximum of 64 persons) as the basic unit for computation. Computation includes the net pay of each individual, the analysis of the number of bills of different denominations, and computations of subtotals of each small group. After computations for a small group have been completed, the total command (P) can be used to total all the subtotals for the shop (scientific laboratory) and place the totals in each memory unit. After computations for a shop (scientific laboratory) have been completed, the accumulation command (L) can be used to accumulate the subtotals of the shops (scientific laboratories) for the entire plant and place the totals in each memory unit. When information on a new shift group or when shop changes are input, the previous totals or accumulated totals are not affected. Each accumulated unit uses a double word length form which can record a maximum of 10^8 yuan, sufficient for all units to be used for 100,000 persons.

The method of deductions for sick and business leave takes into consideration the various different rules and regulations. To satisfy demands of various types of users, the program provides three deduction commands for day, time and minute (T,U,V) and sick leave sign command for users to use. The sick leave sign command is represented by I, the two numbers following I represent the percentage of deduction from the wage. When computing wages with sick and business leave deductions, the internal code and the sick leave sign are input into the machine (those without the sign are treated as business leave) and the day, time and minutes. The program will automatically calculate the amount of deductions and will fill the information in the corresponding deductions unit (K). For example, worker Yi has an internal code 015, he takes 3 and a half days of sick leave and the percentage of deduction from his wages is 30 percent. Thus the string of commands is:

A15*I30*T3*U4*

The program has a remarks column for the user to specify the unit, the number of people in the group, and the date of issuance of the salary. When printing the inventory sheet or punching the paper tape, the remarks column is also output together for the user to review or to keep on file. When necessary, the user can revise the content of the remarks column via the keyboard.

The program has multiple printing functions. On a definite command, it can output lists of wages of the individuals, groups, shops (scientific laboratories) and the entire plant.

The output equipment for printing the lists can be selected from among the teletype and wide line printers which the program is suited.

To adapt to the need of changes in personnel, the program provides means to add or to delete certain personnel.

In addition, the program also has a "user's window" for the user to revise certain units at any time to facilitate control of the output format, change various computational methods to adapt to the needs of the unit.

The "303-1 general purpose wage program" includes a high performance error checking system, thus assuring a high reliability of the program. The program can automatically and vigorously check every operation of the operator. When the operator inputs an item of a wage parameter, the program will automatically check to see if the "certain person" of that parameter exists. If it is discovered that the code (H) representing that person is zero, the program will show an error sign to indicate there is no such person and will refuse to accept that parameter. When the operator sends a computational command or a keypunch command, the program will automatically check the number of people to see if the actual number of people is the same as the number of people listed in the remark column. If the answer is negative, the program will print out "the number of people is incorrect" as an error indicator and will refuse to execute that order. To prevent errors in the accumulative numbers and the totals due to operating mistakes, the program follows the rule that the accumulation of one unit to the next unit one step above can only be performed once, if an accumulation command is sent as a mistake, the program will indicate "repeated accumulation" as an error and will refuse to execute that operation. In addition, unauthorized commands or irrational demands can all be checked out by the program and will be appropriately processed. Because of this layer by layer check, the program is strongly resistant to interference. No matter how the keyboard is operated, there will be no worry of "inundating the program."

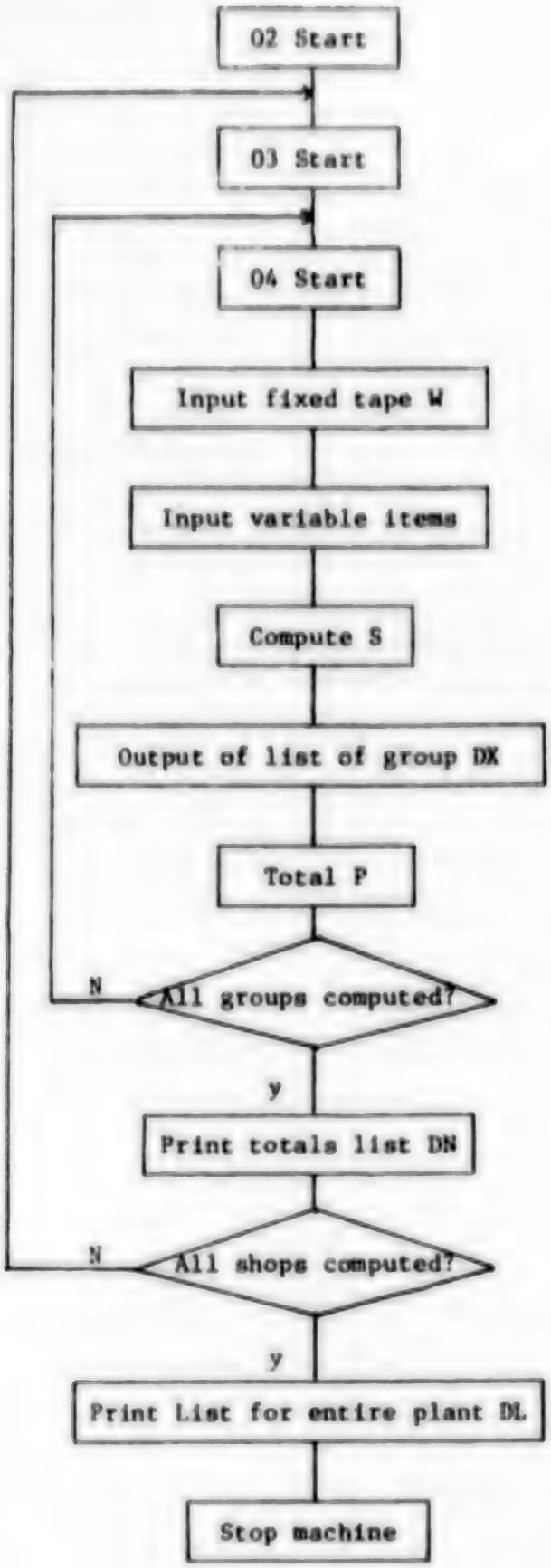
The "303-1 general purpose wage program" is extremely simple to operate and very convenient to use.

The operational flow chart of that program is shown below.

Because the program uses single letters as the names of each of the wage parameters and the commands of each of the parameters for input, and the operation is commanded by single letter commands (except individuals), these commands are easy to remember.

Input of the B type fixed tape provides the operator great convenience and at the same time one paper tape can be used like a disc or magnetic tape. When the operator sends one input command (W), the program will read in a small group of data.

The program has only 30 commands and financial and accounting personnel with only a junior middle school education can operate it after 1 or 2 hours of instruction.


02 Start
03 Start
04 Start
Input fixed tape W
Input variable items
Compute S
Output of list of group DX
Total P
N
All groups computed?
y
Print totals list DN
N
All shops computed?
y
Print List for entire plant DL
Stop machine

Appendix 1 "303-1 General Purpose Wage Program" Commands

Order Number	Commands	Meaning
1	A	Internal code, order number
2	H	Worker number
3	G	Basic wage
4	J	Added wage
5	X	Cleaning cost
6	E	Transportation cost
7	F	Room cost
8	B	Nursery fee
9	C	Meal ticket
10	N	Heating cost
11	Y	Medical cost
12	K	Deductions
13	Q	Others
14	S	Computation
15	P	Total
16	L	Accumulation
17	I	Sick leave sign
18	T	Number of days
19	U	Number of hours
20	V	Number of minutes
21	R	Input A type tape
22	W	Input B type tape
23	ZA	Key A type fixed tape
24	ZB	Key B type fixed tape
25	D1	Print wage table for one worker
26	DX	Print list of one group
27	DH	Print total for shops
28	DL	Print totals for entire plant
29	O	Delete
30	▲	Remark

9296
CSO: 8111 /0788

DJS130-PZ5 INTERFACE OPERATION DETAILED

Shanghai DIANZIJISHU [ELECTRONICS TECHNOLOGY] in Chinese No 3, 20 Mar 80 pp 2.-26

[Article by Fu Shouchun [0126 1108 2797] and Gu Haoran [7357 3185 3544]: "The DJS130-PZ5 Interface"]

[Text] Ultrahigh frequency testing equipment frequently uses the wave detector or the transducer to transform high frequency electromagnetic energy into low frequency electrical voltage or current, and after magnification, its amplitude is shown by various indicators. Then, these values and other parameters are entered into the corresponding computational formulas for computation of the various characteristics parameters in the ultrahigh frequency transmission, generation and reception systems. In operation, there are two important steps: One is the accurate measurement of the amplitude and the other is the complex computations.

The way to solve the first problem is usually by the use of numerical meters to replace the measuring equipment such as meter heads or light indicators commonly used in the past. The way to solve the latter problem is to use the computer. In general, the small computer is sufficient. But to use these two techniques in isolation, the only way is to manually record the results of each measurement and then the experimental results are written into a program. Via special computer equipment, the program is keyed into the computer on the keyboard of punched on cards, and then input into the computer for computations. This way has a higher accuracy from the viewpoint of accuracy but there is no "real time" operation. Man has to be the intermediary in the relationship between the numerical meters and the computer. Can the human intermediary be eliminated by adding an appropriate interface circuit so that the computer and the testing equipment can be connected? We tried. Some interface circuits (not standard interface) were added between a numerical voltmeter PZ5 and the small computer DJS-130 so that the values measured by the numerical voltmeter could be directly input into the computer. We standardized the device code of the numerical voltmeter and the tele-input device. When operating with a BASIC interpretive program, the values of the voltmeter can be input using INPUT statements of BASIC language, and automatically start the computational program to perform computation. Then, the computational result is output. In this way, an "open loop" semiautomatic measurement as shown in Figure 1 is realized. This has established a foundation for future realization of the "closed loop" fully automatic system (as shown by the dotted line).

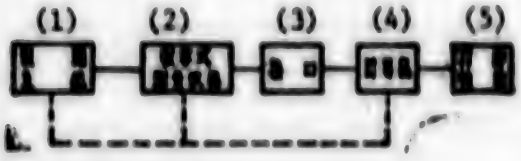


Figure 1

Key:
(1) tested system
(2) digital measuring instrument
(3) interface
(4) computer
(5) printout

Because the interface utilizes the original BASIC language system, none of its interpretive programs has to be added or modified. When inputting data into the computer, the interface is similar to the original tele-input device. To the computer, it is a peripheral equipment equivalent to the tele-input device, therefore the device code of the interface is also "10."

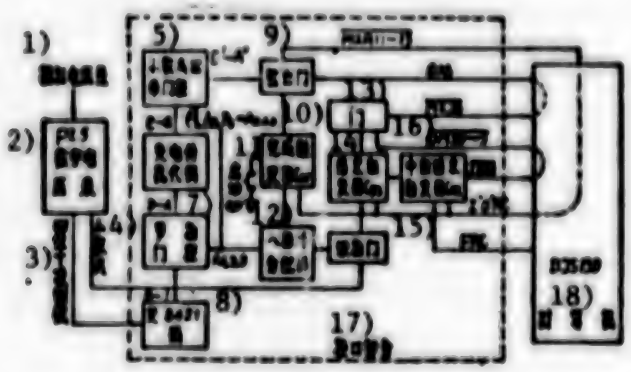
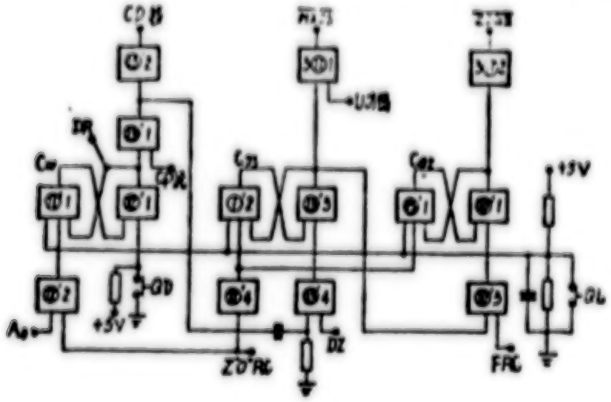


Figure 2

Key:
(1) analog voltage
(2) PZ5 digital voltmeter
(3) four-digit decimal number
(4) decimal point
(5) decimal point carriage return gate control
(6) changed to teletype codes
(7) beat gate control
(8) changed to 8,4,2,1 codes
(9) output gate
(10) completion trigger C_W
(11) start
(12) eight-pulse distributor
(13) gate
(14) termination trigger C_{JS}
(15) missing beat gate
(16) interrupt request trigger C_{QZ}
(17) interface part
(18) computer

The interface operates according to the logic shown in Figure 2. The part of the dotted lines in the diagram is the interface part. As shown in the diagram, the analog voltage measured is first converted to numerical amounts by the numerical amounts by the numerical voltmeter. When the data on the output line of the numerical voltmeter are sent into the computer, pressing the start button QD on the interface will send this group of numbers digit by digit by the process which the tele-input device uses to enter the numbers. The data output by the numerical voltmeter are a group of nonstandardized decimal numbers. To facilitate test manufacture and explanation, the four digits of the effective number of the numerical voltmeter are taken into consideration, including the three digits after the decimal point which may possibly occur together with the carriage return signal as a completely transmitted "word," thus, sending one word of the data requires eight beats time. The eight beats are generated by the eight pulse distributor. The eight pulse distributor is controlled by the completion trigger C_W. It consists of the pulse source CP_{source}, D_1-D_4 shifters and decoder.



When the data of the test are input into the computer, press QD (see Figure 3), set the trigger C_W composed of gate (11)'1 and gate (12)'1 at "1", open gate (13)'1, the pulse from the pulse source CP_{source} passes through gate (13)'1. After passing through gate (13)'2 and the phase is reversed, the pulse enters the shifter via the CP_{shift} serving as a shifter (the shifter is set at full "0" by DR when C_W is cleared), the first pulse only shifts $\overline{D_4}$ = 1 of D_4 into D_1, the carry 4 pulse shifts to full "1", the recarry 4 pulse again shifts to full "0". The decoder decodes the eight beats A_1 - A_0. After the eighth beat A_0 occurs, and when the clear signal of the computer Z"0"RC arrives, C_W is again set at "0" via the gate (12)'2, completing an eight beat cycle.

The first beat sends the first digit, the second beat sends the decimal point, the third beat sends the second digit..... this means, the 1st, 3rd, 5th, 7th beats send the effective digits, the 2nd, 4th, 6th beats send the decimal point, the 8th beat sends the carriage return code. But according to the actual situation of the decimal point of the 2nd, 4th and 6th beats, there is only one beat,

and the other two beats do not exist, therefore a signal to interrupt the sending of digits must be requested from the computer:

$$Zd = A_0 + A_1 + A_3 + A_5 + A_7 + A_2S_1 + A_4S_2 + A_6S_3$$

$$= \overline{\overline{A_2 + A_4 + A_6}} + A_2S_1 + A_4S_2 + A_6S_3$$

$$= \bar{A}_2\bar{A}_4\bar{A}_6 + A_2S_1 + A_4S_2 + A_6S_3$$

The S_1, S_2, S_3 are the signals of the position of decimal points sent by the numerical voltmeter. The in-between time of each beat for which ZD exists is used by gate (13)'4 to control the trigger C_{JS} to set C_{JS}, when the computer's reference beat pulse FRC of the input and output equipment arrives (on each computational beat, the computer always sends out this pulse), it sets C_{QZ} via gate (10)'3, and it is connected to the bus $\overline{ZDQQ}$ of the computer via OC gate 3(11)2 to request the computer for interruption. After the computer receives interrupt request signal, it sends the radix-minus-one complement of the device code via the bus $\overline{DMSO-5}$. When it reaches the (10) code, after it coincides with the decoding circuit of the (10) code set up in the interface, it is connected to the bus $\overline{MXJS}$ via the OC gate 3(11)1 controlled by C_{JS}, and informs the computer that the selection of equipment selected has been completed. After the computer receives this signal, it sends the pulse QAS to access the numbers in the A buffer register. The pulse goes through the gate 3(15)3 already opened by C_w, and finally via data input into the bus $\overline{MXR11-15}$, the numbers of the numerical voltmeter (equivalent to the A register) are decoded and then sent into the computer.

Because the output of the numerical voltmeter is not standard, each digit of the decimal system is output on 10 lines. When -12 volts occurs on a certain line, each represents 0,1,....,9 in order. Thus these 10 states (taking 9 is sufficient) must first be transformed in electrical level and then into 8,4,2,1 codes. The gate control of the electrical potential of the beat is added to form D,C,B,A signals e,d,c,b,a of the teletypewriter (The 10 states can be directly decoded into the e,d,c,b,a signals, the beat control is added in front of the e,d,c,b,a gates).

Also because the 2,4,6 beats may possibly transmit the decimal point, and the 8th beat transmits the carriage return signal, therefore the carriage return signal is A_o, the control signal of the decimal point is $D_s = A_2 + A_4 + A_6 = \bar{A}_2\bar{A}_4\bar{A}_6$. At the same time, one must notice that when transmitting the carriage return signal and the decimal point signal, e,d,c,b,a are equivalently transmitted as "O", and the corresponding code is "01101". Thus, the signals of the effective numerical codes, the signals of the decimal point, the carriage return signal are finally combined to form the signals E',D',C',B',A'. These signals are connected to the data bus $\overline{MXR11-15}$ of the computer by the OC gate control. After the computer receives the data, the interrupt control program sends the signal Z"O"RC to the equipment and clears C_w, C_{JS} and C_{QZ}. Thus, the entire process of input of data in PZ5 into the DJS-130 computer is realized.

For example, when we measure stationary wave ratios, we only need to connect the direct current signal voltage sent out by the wave detector of the measuring line to the interface system of the numerical voltmeter and computer, and after the computer "begins operation," the BASIC interpretive program is input, the following program is input and the stationary wave ratio can be measured and computed.

```
10      INPUT       N
20      PRINT
30      INPUT       V1, V2
40      PRINT
50      LET         Z = SQR(V1/V2)
60      PRINT       'Z = 'Z
70      GOTO        20
        RUN
```

After this program has been input, the computer executes the program, when it reaches the 10th statement, it prints out a "#" symbol, requests the N value, and dynamically stops the machine. At this time, we must first press the "1...." key on the typewriter, then press the D button on the interface. The typewriter types out a number following the "#" (showing the data can be entered into the computer.....) and changes line and types out the "#" symbol to request the numerical value of V1. At this time, the adjustable measuring line can separately measure the greatest and least values. The D is restarted and after two numbers have been input, the computer immediately computes and prints out the computed results Z = ********.

If measurements need to be taken several times, the measuring line can be again the greatest and least values and the QD can be restarted, and the results can be printed out again,

```
Z = ********

.............
```

After actual experiment, the use of this interface system to conduct measurements and computations is shown to be very convenient.

9296
CSO: 8111/0787

END

END OF FICHE DATE FILMED

13 Jan. 1981

D.D.